Reni Urquiaga ENTD278

**IT Online Training Application Design Specification**

# 1. Introduction

The following document will outline the process for designing and implementing the application process of the system. Diagrams are created to show the functionality of the system through the application and how it interacts with the various actors.

## 1.1 Purpose

The purpose of the document is to present how the Non-Functional requirements of ITOT can be met through the use of the application. Use Cases will be used as

## 1.2 Scope

The software being built will be able to meet the functional and non-functional requirements of the client, it will be designed with a Client/Server design where the system will be split into two applications, with the Users of the System using remote machines to make requests to the application on the server that hosts a database with an application to exchange information.

## 1.3 References

The IT Online Training (ITOT) Analysis Specification document will be used to reference the requirements and other information that was researched for this project.

# 2. System Architecture

A good architectural design is important to meet the needs of the client and reduce the downtime of the system for routine or unscheduled maintenance. The section provides an outline for the system architecture being designed for ITOT. The system will utilize a Client/Server architectural style to meet the demands of the client.

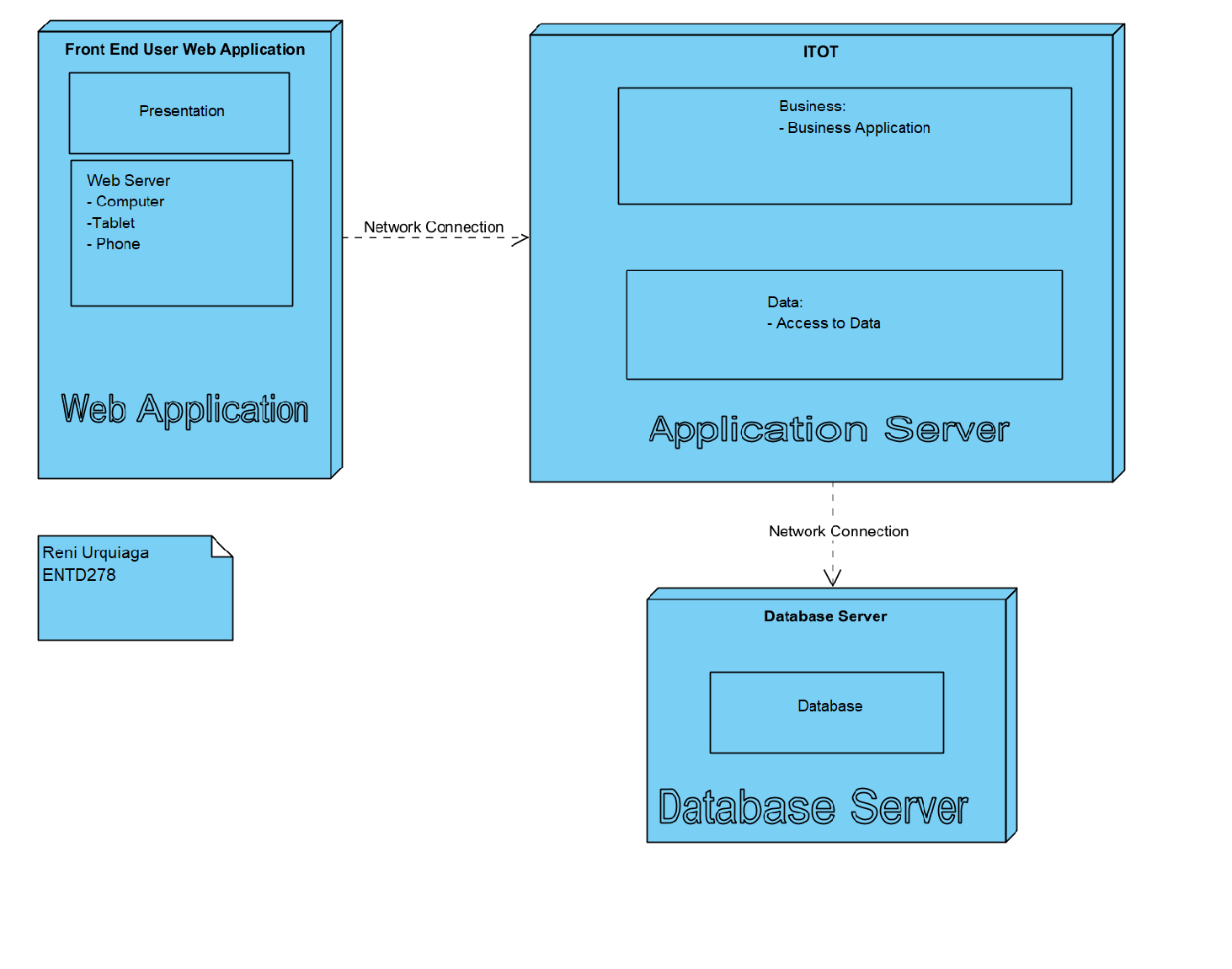
## 2.1. Assumptions and Dependencies

The User will require hardware such as a computer, tablet or phone that will allow them to run a web based interface that will allow them to interface with the system in order to send and receive information. The Server-side of the system will require a a database with storage capabilities and the ability to run software that will be developed in order to make requests to the database and allow modifications to products and other data classes if necessary. The system should also allow security to prevent access to sensitive information and other types of attacks that can be presented from bad actors.

## 2.2. General Constraints

The major constraint will be the ability to develop an application for the client with the appropriate features that they are requesting. This will require a lot of resources in order to design from the ground up. Designing an application that can scale with the clients demands while requiring little change to the entire system will be a priority to the design of the application. Furthermore, providing security to the system will be vital in order to protect information and keep intruders out.

## 2.3 Proposed System Architecture

**

### 2.3.1 System Architecture Discussion

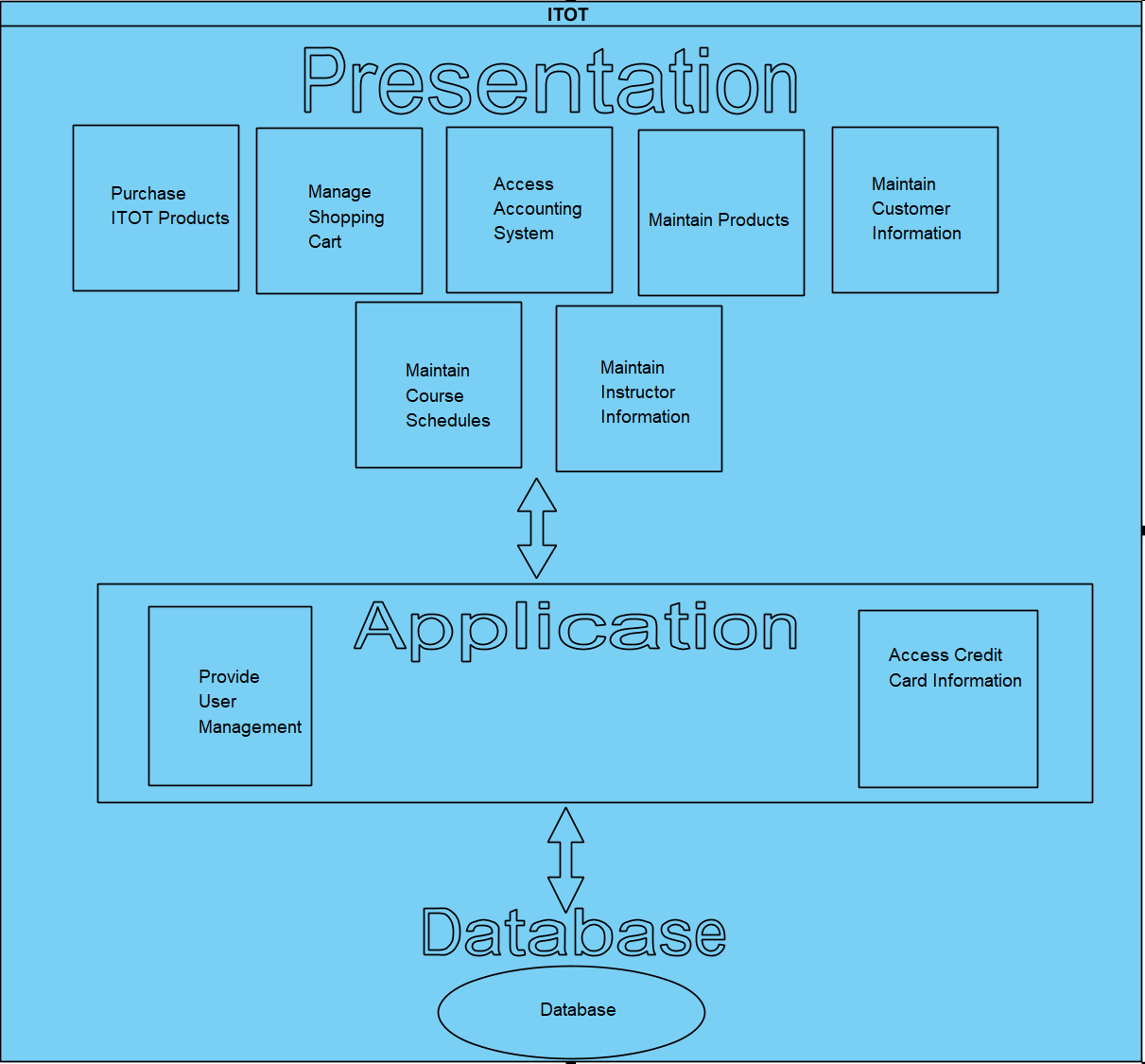
The system architecture composes of a Presentation Layer where a web application will be accessible through a user’s device such as a computer, tablet or phone. Customers and Administrators will use forms on the web application to search for and modify products as well as purchase products through a connection to a business application on the Application server. This server will reside on the business side and will allow for all the processing and programming required to build the ITOT system. The Application server will also allow for access to data from a Database Server which will contain data for the system, including sensitive information. The Application Server will allow for scalability while the Database layer will allow the system to utilize security for the data.

# 3. 4+1 Architecture

The following diagrams will represent different aspects of the application design. The diagrams will include a Logical and Implementation Views for the functional portions of the application and Process and Deployment View for the Non-Functional portions of the application.

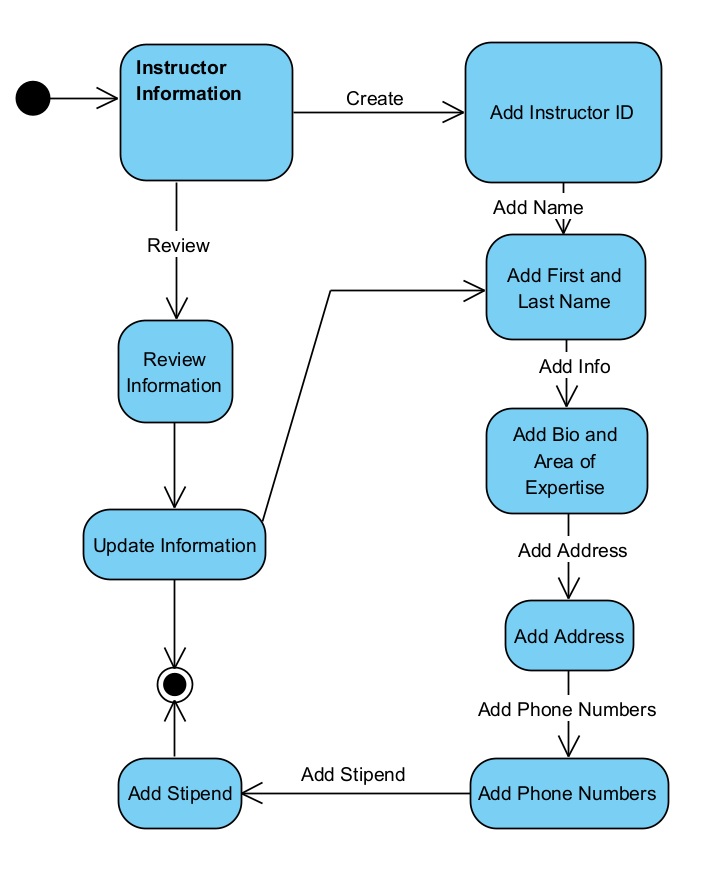
## 3.1 Logical View

The Logical View, intended for Analysts and Designers to illustrate how the application system will be structured. The functions in the Presentation zone of the diagram will allow users to perform them on the user end via a capable device, while the functions on the Application end will be performed on the System Application and be pushed from there. The Database will store data that will be shared through the business zone and onto users systems when applicable.



## 3.2 Process View

The Process View is intended for system integrators during the design stage and focuses on performance, scalability and throughput. The following diagram will show how the system will behave when performing the function of Maintaining Instructor Information. The system takes inputs from the Instructor user in order to create or modify an Instructor profile.

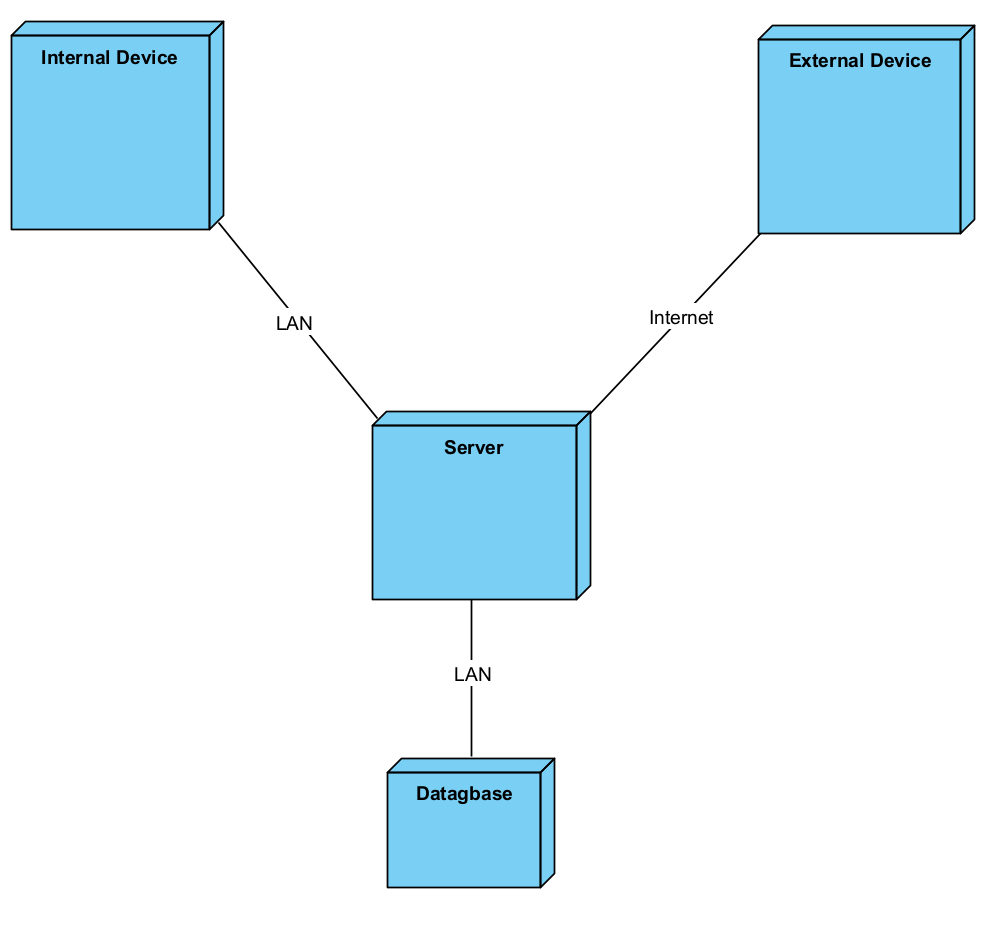
**

## 3.3 Implementation or Development View (Subsystem Decomposition)

*<According to FCGSS (2007), “this is a view of a system’s architecture that encompasses the components used to assemble and release a physical system. This view focuses on configuration management and actual software module organization in the development environment. The software is actually packaged into components that can be developed and tested by the development team”.* ***Include a brief description of the UML diagram you would propose for this view. If you have already created this model, insert it here.****>*

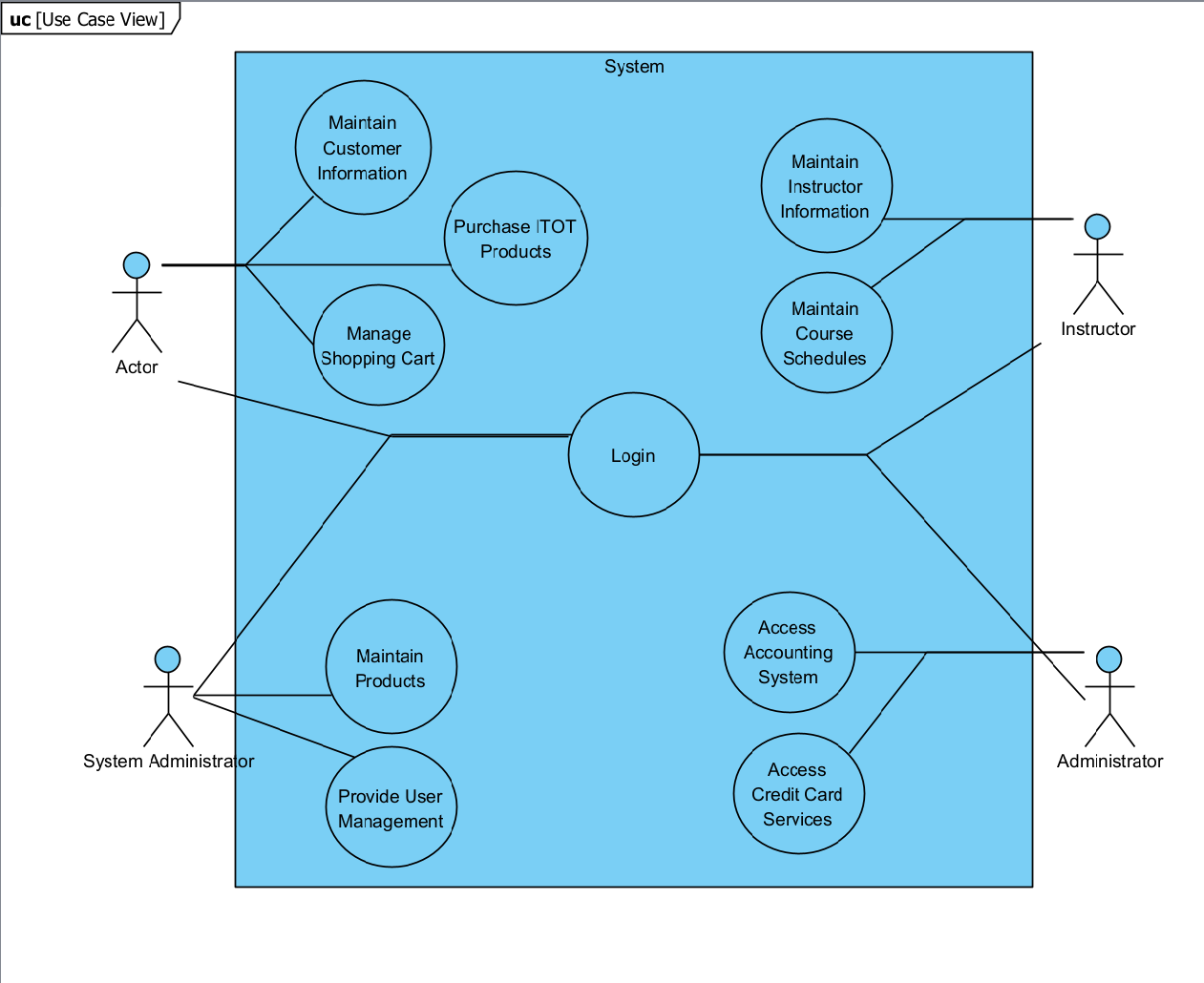
## 3.4 Deployment View

The Deployment View used by System Engineers shows the areas where hardware and software should be used. The diagrams shows that Internal or External Devices will access the application which resides on a server. Internal devices will communicate through LAN while External will use the internet. The Server will access the database through LAN as well.



## 3.5 Use Case View

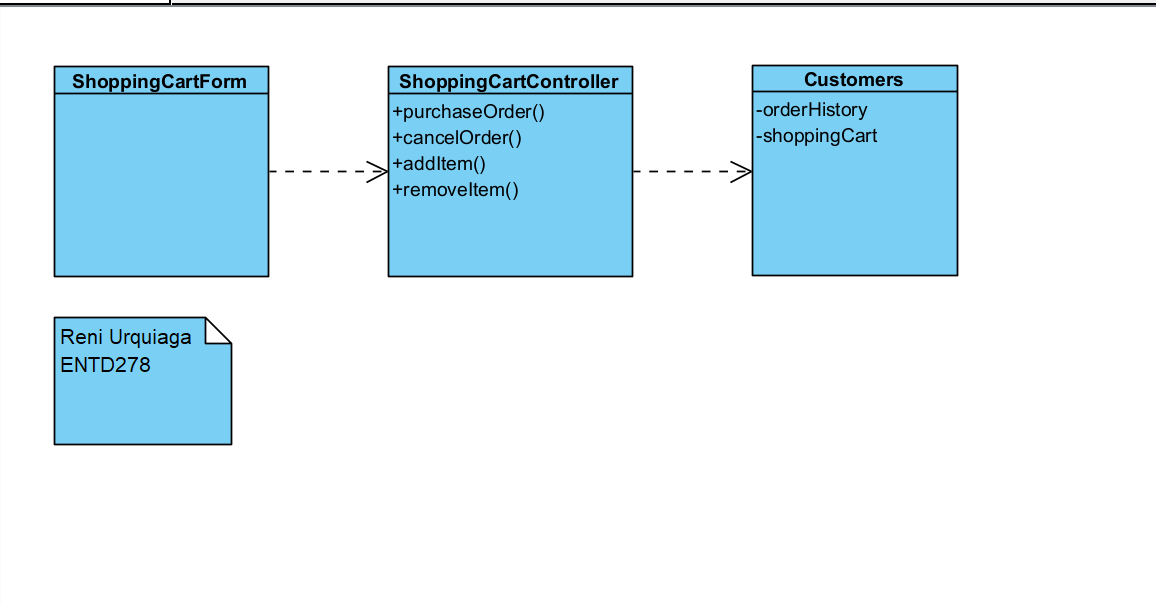
The Use Case View illustrates how users will interact with the system. The following use case is very generic, outlining which actors will interact with certain functional areas of the system. It also illustrates what the system will be capable of once the application is fully designed and deployed.



# 4. Application Class Design

The following Application Class Diagram will represent the ShoppingCart class and its interaction for the Manage Shopping Cart Use Case.

## 4.1 Application Class Diagram



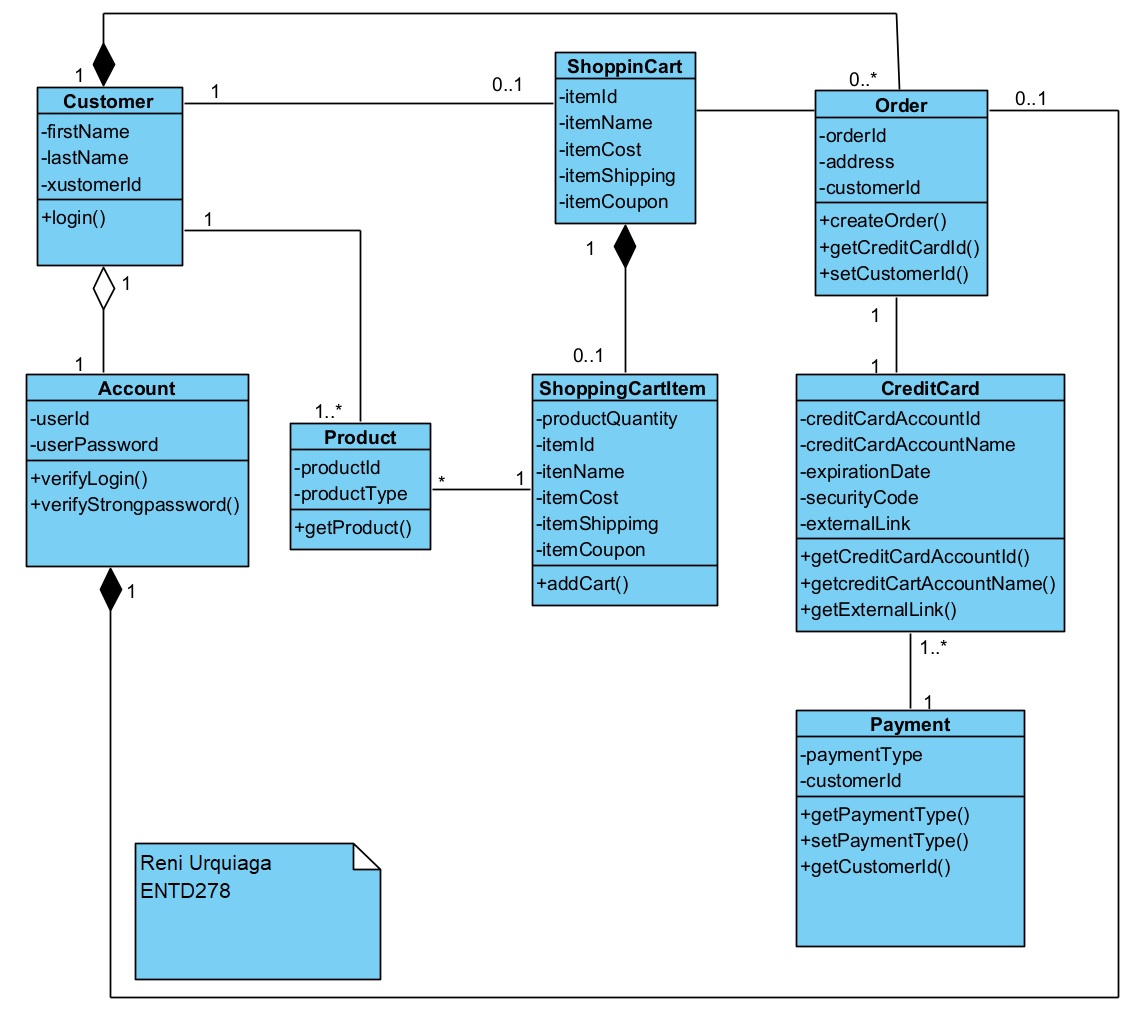
## 4.2 Application Class Diagram Discussion

The Application Class Diagram represents the detailed use case for the Shopping Cart feature. In the diagram the ShoppingCartForm is used as the Boundary, ShoppingCartController as the Controller and Customer as the Entity. The modeling illustrates the use of a controller to mediate the flow of information between the boundary and entity, which allows for updating the interface of the form without disrupting the application.

# 5. Class Design

The following detailed class diagram will show the relationships and associations for the ShoppingCart and ShoppingCartItem classes.

## 5.1. Detailed Class Diagram

**

## 5.2. Detailed Class Diagram Discussion

The above diagram shows the detailed process for the ShoppingCartItem and ShoppingCart use case. The diagram shows the attributes and the operations for the use case, the getter/setters are based on the operation being performed of a Customer searching for an Product and adding it to a cart. The product is issued a ShoppingCartItem ID while in the cart and the next step is the ordering process and assigning the Order to the customer once completed and choosing a payment method for the order. The diagram also utilizes Aggregation and Composition lines to illustrate elements that die or live on.

# 6. Testing

The following section provides two Test Case Scenarios for ShoppingCart Management. One Test Case Scenario is for adding an item into the Shopping Cart and the other is for removing an item from the Shopping Cart. Both Test Case Scenarios lack an Actual Result and Summary due to not having an actual testing environment where the scenarios can be tested, therefore those cells are left blank.

## 6.1 Test Case Scenario

**Test Scenario Name:** Manage Shopping Cart

**Test Scenario Description:** Steps for adding an item to a Shopping Cart.

## 6.2 Test Case 1

Manage Shopping Cart – Add Item

| **Test Case 1 Name: Add Shopping Cart Item** | | | | |
| --- | --- | --- | --- | --- |
| **Test Case 1 Description:** This test case shows the steps to add an item to a Shopping Cart. | | | | |
| **Test Case 1 Prerequisites:** User must be logged in to account. User must search for a product and be on item page. | | | | |
| **Test Case 1 Steps** | **Test Case 1 Input** | **Test Case 1 Expected Result** | **Test Case 1 Actual Result** | **Test Case 1 Status** |
| Select Product | User selects product and adds desired quantity by clicking “Add to Cart” on product page. | System adds the item and quantity to Shopping Cart. |  |  |
| View Cart | User selects Shopping Cart link to confirm that items were added to cart. | Shopping Cart shows correct product and quantity on Shopping Cart page. |  |  |
| Out of Stock Item | User selects an item that is out of stock. | System displays error indicating that product is out of stock. |  |  |

*>*

### 6.2.1 Test Case 1 Discussion

The above test case scenario is for adding an item into a Shopping Cart. The prerequisites are that the User is logged into their account and they are on a product page. The first step is to select a product with the expected result being that the item is added to the shopping cart. The second step is for the user to view the cart and the result being the shopping cart should display the item and quantity. If the user selects an item that is out of stock, the result will be the system displaying an error that the product is out of stock and no change is made to the shopping cart.

6.3 Test Case 2

Manage Shopping Cart – Remove Item

| **Test Case 2 Name: Remove Shopping Cart Item** | | | | |
| --- | --- | --- | --- | --- |
| **Test Case 2 Description:** This test case shows the steps to remove an item to a Shopping Cart. | | | | |
| **Test Case 2 Prerequisites:** User must be logged in to account. User must be on shopping cart page. | | | | |
| **Test Case 2 Steps** | **Test Case 2 Input** | **Test Case 2 Expected Result** | **Test Case 2 Actual Result** | **Test Case 2 Status** |
| Remove Item | Select icon on page to remove item from Shopping Cart. | System removes item from Shopping Cart. |  |  |
| Quantity Updated | View Updated Quantity | System confirms User has made this choice. If yes: Page refreshes to update Shopping Cart quantity.  If No: Message clears and no Items are removed. |  |  |
| Clear Shopping Cart | User clicks on icon to clear all contents from Shopping Cart. | System confirms User has made this choice.  If yes: Page refreshes to update All Items removed from Shopping Cart  If No: Message clears and no Items are removed. |  |  |

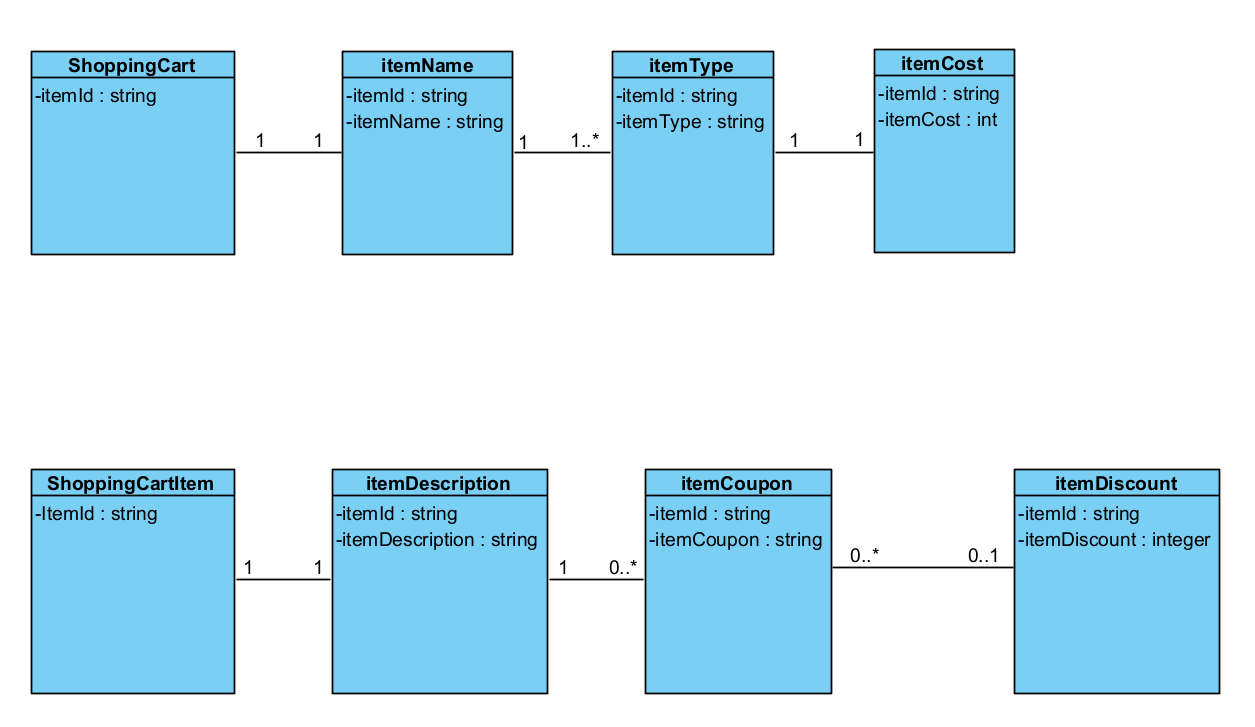
### 6.3.1 Test Case 2 Discussion

The above test case scenario is for remove an item into a Shopping Cart. The prerequisites are that the User is logged into their account and they are on the Shopping Cart page. The first step is to remove the item by selecting the icon on the page to remove the undesired item from the cart. Prior to removing the item, the user will encounter a message indicating whether or not they want to remove the item and they must confirm the decision to remove the item. The System is expected to remove the item and update the quantity. The user can view the updated quantity after the page is refreshed. The user also has the option to clear the entire cart and will have to confirm their choice with the same message and the entire contents of the shopping cart will be cleared if “yes” is selected and no changes is “no” is selected.

# 7. Data Design

The following section illustrates the data design between the ShoppingCart and ShoppingCartItems and how they will be stored and related within a database.

## 7.1 Persistent Data Model

**

### 7.1.2 Persistent Data Model Discussion

The diagram above illustrates the persistent data for the ShoppingCart and ShoppingCartItem.

The ShoppingCart diagram, the ”ShoppingCart” class is represented with an itemId attribute and is the Primary Key. This attribute has a 1 to 1 relationship with itemName and the same itemId along with an itemName attribute. The itemName class has a 1 to many relationship with itemType, as an itemId can have more than one itemType. The itemType has an itemCost and the relationship is 1 to 1. All classes share the “itemId” as a Primary Key.

The second diagram illustrates the ShoppingCartItem class with an itemId as well as an attribute which again will serve as a Primary Key. The ShoppingCartItem has a 1 to 1 relationship with itemDescription, as an item can only have one description. The itemId has a 1 to 0 or many relationship with itemCoupon, the next class in the diagram, as it can have 1 or 0 coupons but not more than one. The final class in the diagram is itemDiscount which is applied by the coupon and it has a 0 or none to 0 or 1 discount for the item, which is represented as an attribute with integer quality

# Appendix.